Buffered Clock Tree Synthesis with Optimal Clock Skew Scheduling for Reduced Sensitivity to Process Parameter Variations

José Luis Neves and Eby G. Friedman

Department of Electrical Engineering
University of Rochester
Rochester, NY 14618
neves,friedman@ee.rochester.edu

Abstract - An integrated top-down design system is presented in this paper for synthesizing clock distribution networks for application to synchronous digital systems. The timing behavior of a digital system is considered at the register transfer level, permitting a non-zero clock skew schedule to be determined which reduces the clock period as compared to zero skew-based approaches. The concept of a *permissible range* of clock skew for each local data path is introduced. The choice of clock skews considers several design objectives, such as minimizing the effects of process parameter variations, imposing a zero clock skew constraint among the input and output registers, and constraining the permissible range of each local data path to a minimum value.

The clock path delays and the worst case variation of the primary process parameters are used to determine the hierarchical topology of the clock distribution network, defining the number of levels and branches of the clock tree and the delay associated with each branch. The delay of each branch of the clock tree is physically implemented with distributed buffers targeted in CMOS technology using a circuit model that integrates short-channel devices with the signal waveform shape and the characteristics of the clock tree interconnect. A bottom-up approach for calculating the worst case variation of the clock skew due to process parameter variations is integrated with the top-down synthesis system. Thus, the local clock skews and a clock distribution network are obtained which are less sensitive to process parameter variations.

This methodology and related algorithms have been demonstrated on several MCNC/ISCAS-89 benchmark circuits. Increases in system-wide clock frequency of up to 50% as compared with zero clock skew implementations are shown. Furthermore, examples of clock distribution networks that exploit intentional localized clock skew are presented which are insensitive to process parameter variations with worst case clock skew variations of up to 30%.

1. Introduction

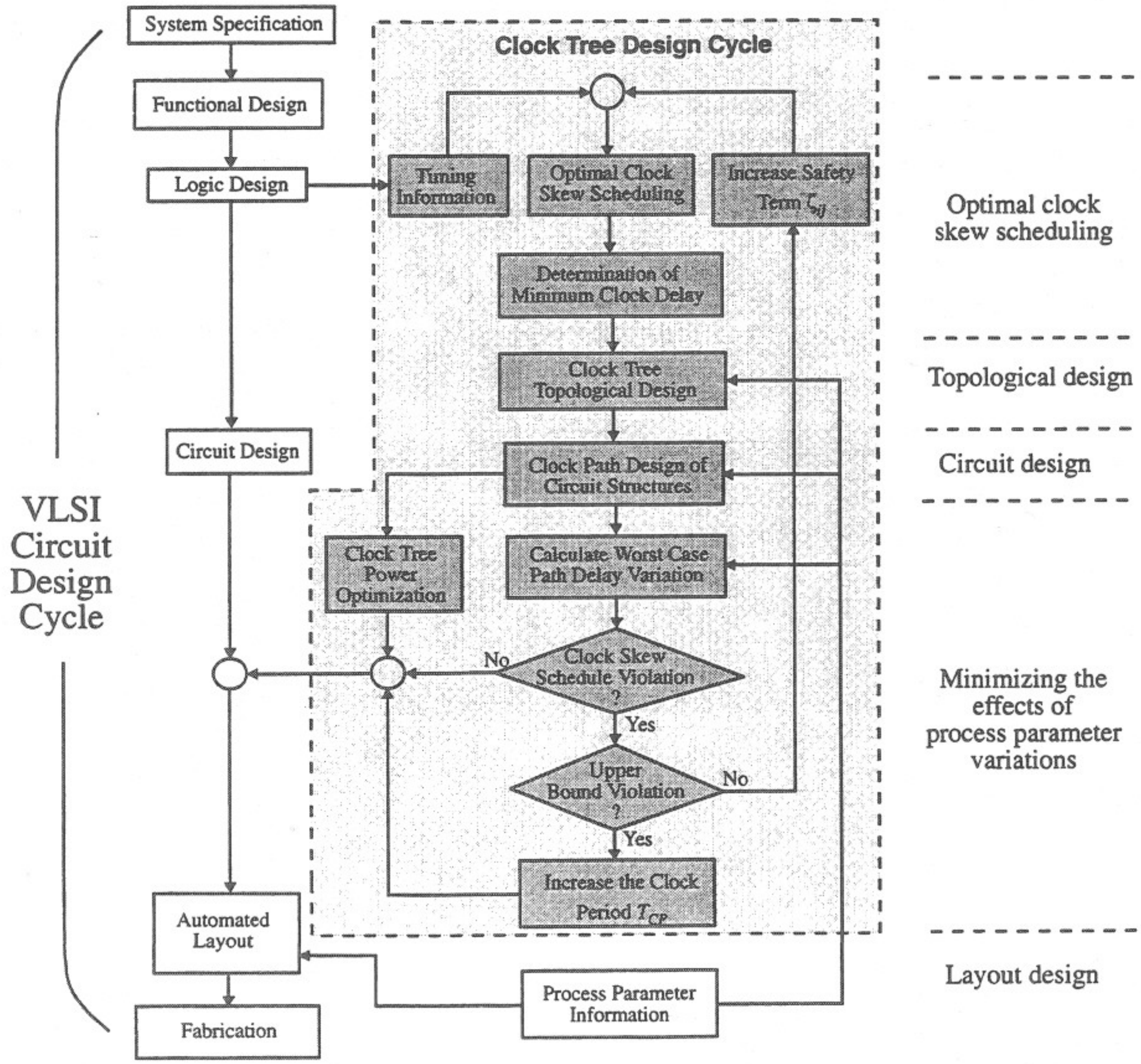
Most existing digital systems utilize fully synchronous timing, requiring a reference signal to control the temporal sequence of operations. Globally distributed signals, such as clock signals, are used to provide this synchronous time reference. These signals can dominate and limit the performance of VLSI-based digital systems. The importance of these global signals is, in part, due to the continuing reduction of feature size concurrent with increasing chip dimensions. Thus interconnect delay has become increasingly significant, perhaps of greater importance than active device delay. The increased global interconnect delay also leads to significant differences in clock signal propagation within the clock distribution network, called *clock skew*, which occurs when the clock signals arrive at the storage elements at different times. The clock skew can be further increased by unintentional factors such as process parameter variations which may limit the maximum frequency of operation, as well as create race conditions independent of clock frequency, leading to circuit failure. Therefore, the design of high performance, process insensitive clock distribution networks is a critical phase in the synthesis of synchronous VLSI digital circuits. Furthermore, the design of the clock distribution network, particularly in high speed applications, requires significant amounts of time, inconsistent with the high turnaround in the design of the more common data flow elements of digital VLSI circuits.

Several techniques have been developed to improve the performance and design efficiency of clock distribution networks, such as placing distributed buffers within clock tree layouts [1] to control the propagation

This research is based upon work supported by Grant 200484/89.3 from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil), the National Science Foundation under Grant No. MIP-9208165 and Grant No. MIP-9423886, the Army Research Office under Grant No. DAAH04-93-G-0323, and by a grant from the Xerox Corporation.

delay and power consumption characteristics of the clock distribution networks, using symmetric distribution networks, such as H-tree structures [2], to minimize clock skew, and applying zero-skew clock routing algorithms [e.g., 3,4] to the automated layout of high speed clock distribution networks in cell-based circuits. Effort has also been placed on reducing clock skew due to process variations [e.g., 5, 6], and on designing clock distribution networks so as to ensure minimal variation in clock skew [1, 4]. Alternative approaches have been developed for using intentional non-zero clock skew to improve circuit performance and reliability by properly choosing the local clock skews [6-8]. Targeting non-zero local clock skew, a synthesis methodology has been developed for designing clock distribution networks capable of accurately producing specific clock path delays [9, 10]. These clock distribution networks exploit intentional localized clock skew and, as is described in this paper, consider the effects of process parameter variations.

A design environment is presented in this paper for efficiently synthesizing distributed buffer, tree-structured clock distribution networks. This methodology is illustrated in terms of the IC design process cycle in Figure 1. The IC design cycle typically begins with the System Specification phase. The Clock Tree Design Cycle utilizes timing information from the Logic Design phase, such as the minimum and maximum delay values of the logic blocks and the registers. Process parameter information is extensively used in several stages of the design environment for ensuring the accuracy of the clock tree. The output of the Clock Tree Design Cycle is a detailed circuit description of the clock distribution network, including the number and geometric size of each buffer stage within each branch of the clock tree.



System Specification
Functional Design
Logic Design
Circuit Design
Automated Layout
Fabrication
VLSI Circuit Design Cycle
Clock Tree Design Cycle
Timing Information
Optimal Clock Skew Scheduling
Increase Safety Term ζ_{ij}
Determination of Minimum Clock Delay
Clock Tree Topological Design
Clock Path Design of Circuit Structures
Clock Tree Power Optimization
Calculate Worst Case Path Delay Variation
Clock Skew Schedule Violation ?
No
Yes
Upper Bound Violation ?
No
Yes
Increase the Clock Period T_{CP}
Process Parameter Information
Optimal clock skew scheduling
Topological design
Circuit design
Minimizing the effects of process parameter variations
Layout design

Figure 1: Block diagram of the clock tree design cycle integrated with standard IC design flow

This paper is organized as follows: in section 2, a localized clock skew schedule is derived from the effective permissible range of the clock skew for each local data path considering any global clock skew constraints and process parameter variations. In section 3, a topology of the clock distribution network is obtained, producing a clock tree with specific delay values assigned to each branch. The design of circuit structures for implementing the individual branch delay values is summarized in section 4. In section 5, techniques for selecting the local clock skew values so as to be less sensitive to process parameter variations are presented. In section 6, these results are evaluated on a series of circuits, thereby demonstrating performance improvements and immunity to process parameter variations. Finally, some conclusions are drawn in section 7.

2. Optimal Clock Skew Scheduling

A synchronous digital circuit C can be modeled as a finite directed multi-graph $G(V,E)$. Each vertex $v_j \in V$ in the graph is associated with a storage element, circuit input, or circuit output. Each edge in the graph, $e_{ij} \in E$, represents a physical connection between vertices v_i and v_j, with an optional combinational logic path between the two vertices. A local data path L_{ij} is a set of two vertices connected by an edge, $L_{ij} = \{v_i, e_{ij}, v_j\}$ for any $v_i, v_j \in V$. A global data path $P_{kl} = v_k \xrightarrow{p} v_l$ is a set of alternating edges and vertices $\{v_k, e_{k1}, v_1, e_{12}, ..., e_{n-1l}, v_l\}$, representing a physical connection between vertices v_k and v_l, respectively. An edge is a bi-weighted connection representing the maximum (minimum) propagation delay T_{PDmax} (T_{PDmin}) between two sequentially adjacent storage elements. A multi-input circuit can be modeled as a single input graph, where each input is connected to vertex v_0 by a zero-weighted edge.

2.1 Timing Constraints

The timing behavior of a local data path L_{ij} is described in terms of the minimum propagation delay through L_{ij} as

$$T_{PD(min)} = T_{C-Qi} + T_{Logic(min)} + T_{Int} + T_{Set-upj} \quad , \qquad (1)$$

and the maximum propagation delay through L_{ij} as

$$T_{PD(max)} = T_{C-Qi} + T_{Logic(max)} + T_{Int} + T_{Set-upj} \quad , \qquad (2)$$

where T_{Skewij} is the clock skew between R_i and R_j, T_{Hold} is the amount of time the input data must be stable once the clock signal changes state, T_{C-Q} is the time required for the data to leave R_i once it is triggered by a clock pulse C_i, $T_{Logic(min)}$ ($T_{Logic(max)}$) is the minimum (maximum) propagation delay through the logic block between the registers R_i and R_j, T_{Int} accounts for the interconnect delay, and T_{Set-up} is the time required to successfully propagate to and latch the data within R_j.

The timing behavior of a circuit C can be described in terms of two sets of timing constraints, local constraints and global constraints. The local constraints are designed to ensure the correct latching of data into the registers of a local data path. In particular, (3) prevents latching the incorrect data signal into R_j by the clock pulse that latched the same data into R_i (creating a race condition),

$$T_{Skew}(L_{ij}) \geq T_{Holdj} - T_{PD(\min)} + \zeta_{ij} \quad , \qquad (3)$$

where ζ_{ij} is a safety term to provide some margin in a local data path against race conditions due to process parameter variations, and (4) guarantees that the data signal latched in R_i is latched into R_j by the following clock pulse,

$$T_{CP} \geq T_{Skew}(L_{ij}) + T_{PD(\max)} \quad . \qquad (4)$$

Constraints (3) and (4) are similar to the synchronous constraints introduced in other works, particularly in [7, 8, 11, 12], where the clock skew $T_{Skewij} = T_{CDi} - T_{CDj}$, where T_{CDi} is the delay of the i^{th} clock path.

Assuming that the minimum and maximum delay of each combinational logic block and register are known, a region of valid clock skew for each local data path, called the **permissible range** $Pl(L_{ij})$, is introduced, as shown in Figure 2. The bounds of $Pl(L_{ij})$ are determined from the local constraints, (3) and (4), for a given clock period T_{CP}. Also, the *width* of a permissible range is defined as the difference between the maximum ($T_{Skewij(max)}$) and the minimum ($T_{Skewij(min)}$) clock skew.

Satisfying the clock skew constraints of each individual local data path does not guarantee that the clock skew between two vertices of a global data path P_{kl} is satisfied, particularly when there are multiple parallel and feedback paths between the two vertices. Since any two registers connected by more than one global data path are each driven by a single clock path, the clock skew between these two registers is unique and the permissible range of every path connecting the two registers must contain this clock skew value to ensure that the circuit will operate correctly.

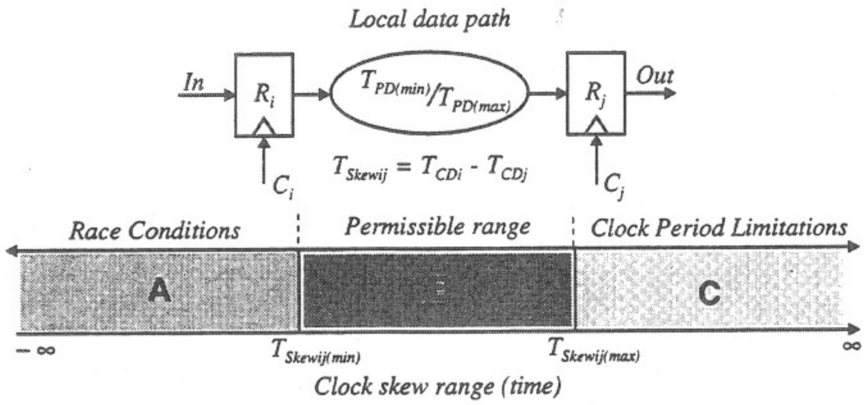


Figure 2: Permissible range of the clock skew of a local data path.

As an example, consider the circuit illustrated in Figure 3, where the numbers assigned to the edges are the maximum and minimum propagation delay of each L_{ij}, and the register set-up and hold times are assumed to be zero. Furthermore, the pair of clock skew values associated with a vertex are the minimum and maximum clock skew calculated with respect to the origin vertex v_0 for a given clock period. The minimum bound of $Pl(L_{ij})$ is given by (3) and is $T_{Skewij(min)} = -T_{PDmin}$ and the maximum bound of $Pl(L_{ij})$ is given by (4) and is $T_{Skewij(max)} = T_{CP} - T_{PDmax}$. Observe that in Figure 3a, a clock period $T_{CP} = 6$ time units (tu) defines a permissible range for each individual local data path. However, no clock skew value exists that is common to the paths connecting vertices v_1 and v_3. The common value for T_{Skew13} is only obtained when the clock period is increased to 8 tu.

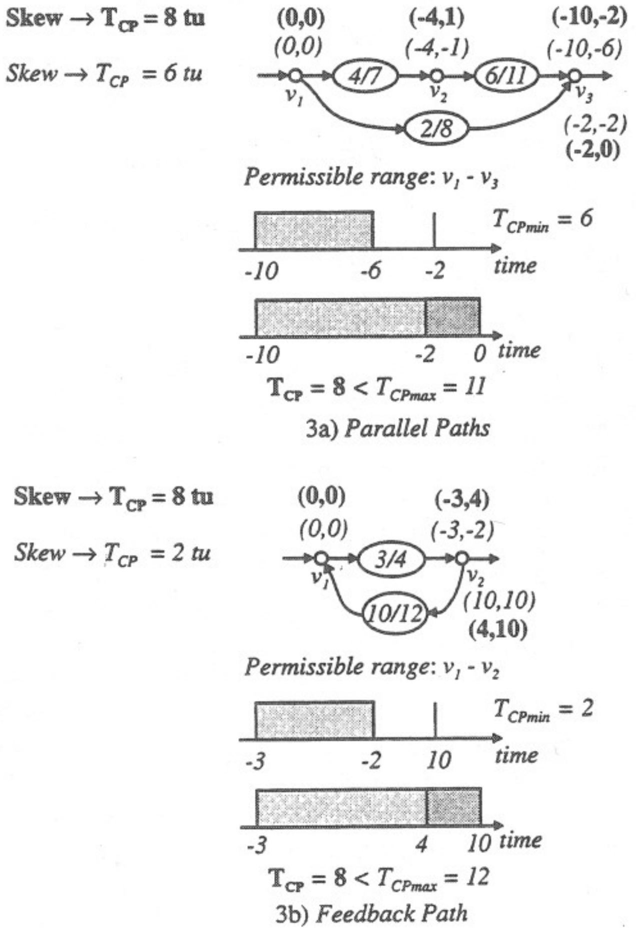


Figure 3: Matching permissible clock skew ranges by adjusting the clock period T_{CP}

To guarantee that a clock skew value exists for any pair of registers within a global data path, a set of global timing constraints must be satisfied. The global timing constraints (5) and (6) are used to calculate the permissible range of any global data path $P_{kl} \in V$, and are based on the permissible range of the local data paths within the respective global data path. In particular, (5) determines the minimum and maximum clock skew of a global data path with respect to v_k, while (6) constrains the clock skew of two vertices connected by a forward and a feedback path. These two constraints can be formally stated as:

Lemma 1: For any global data path $P_{kl} \in V$, clock skew is conserved. Alternatively, the clock skew between *any* two storage elements, v_k, $v_l \in V$, is the sum of the clock skews of each local data path L_{k1}, L_{12}, ..., L_{n-1l}, where L_{k1}, L_{12}, ..., L_{n-1l} are the local data paths within P_{kl},

$$T_{Skew}(P_{kl}) = T_{Skew}(L_{k1}) + T_{Skew}(L_{12}) + \cdots + T_{Skew}(L_{1-1l}) \ . \quad (5)$$

Lemma 2: For any given global data path P_{kl} containing feedback paths, the clock skew in a feedback path between any two storage elements, say v_m and $v_n \in P_{kl}$, is the negative of the clock skew between v_m and v_n in the forward path,

$$T_{Skew}(P_{kl}) = -T_{Skew}(P_{lk}) \quad . \qquad (6)$$

In the presence of parallel and feedback paths, a permissible range only exists if there is overlap among the permissible ranges of each individual parallel and feedback path. Furthermore, the upper and lower bounds of the permissible range are determined from the upper and lower bounds of the permissible ranges of each parallel and feedback path. Formally, the concept of permissible range overlap and the upper and lower bounds of the permissible range of a global data path P_{kl} can be stated as follows:

Lemma 3: Let $P_{kl} \in V$ be a global data path with m parallel and n feedback paths. Let the two vertices, v_k and $v_l \in P_{kl}$, which are not necessarily sequentially adjacent, be the origin and destination of the m parallel and n feedback paths, respectively. Also, let $Pg(P_{kl})$ be the permissible range of the global data path composed of vertices v_k and v_l. $Pg(P_{kl})$ is a non-empty set of values if and only if the intersection of the permissible ranges of each individual parallel and feedback path is a non-empty set, or

$$Pg(P_{kl}) = \left(\bigcap_{i=1}^{m} Pg(P_{kl}^{i}) \right) \cap \left(\bigcap_{j=1}^{n} Pg(P_{lk}^{j}) \right) \neq \varnothing \quad . \qquad (7)$$

Lemma 4: Let the two vertices, v_k and $v_l \in P_{kl}$, be the origin and destination of a global data path with m forward and n feedback paths. If $Pg(P_{kl}) \neq \varnothing$, the upper bound of $Pg(P_{kl})$ is given by

$$T_{Skew}(P_{kl})_{\max} = \min_{1 \le i \le m} (T_{Skew}(P_{kl}^{i})_{\max}), \min_{1 \le j \le n} (T_{Skew}(P_{lk}^{j})_{\min}) \ , \qquad (8)$$

and the lower bound of $Pg(P_{kl})$ is given by

$$T_{Skew}(P_{kl})_{\min} = \max_{1 \le i \le m} (T_{Skew}(P_{kl}^{i})_{\min}), \max_{1 \le j \le n} (T_{Skew}(P_{kl}^{j})_{\max}) \ . \qquad (9)$$

Two global timing constraints impose zero clock skew among the I/O storage elements and limit the permissible clock skew range that can be implemented by the fabrication technology. By constraining the clock skew among the off-chip registers to zero, race conditions are eliminated among all integrated circuits controlled by the same clock source by avoiding the propagation of a non-zero clock skew beyond the integrated circuit. This condition is represented by the following expression,

$$T_{SkewI/O} = T_{SkewI1} + \cdots + T_{Skewn-1n} + T_{SkewnO} = 0 \quad . \qquad (10)$$

An immediate consequence of (10) is that the clock path delay from the clock source to every input and output register is equal.

Although the permissible range of a local data path is theoretically infinite, practical limitations place constraints on the minimum clock path delays that can be implemented with a given fabrication technology. These clock path delays determine the minimum clock skew that can be assigned to any two vertices in the circuit. These fabrication dependent timing constraints are

$$\begin{aligned} \left| T_{Skew}(L_{ij})_{\max} - T_{Skew}(L_{ij})_{\min} \right| &\ge C_1 \quad , \\ \left| T_{Skewij} \right| &\ge C_2 \quad , \end{aligned} \qquad (11)$$

where C_1 and C_2 are dependent on the fabrication technology and are a measure of the statistical variation of the process parameters.

2.2 Optimal Clock Period

Determining an optimal clock period for synchronous circuits while exploiting non-zero clock skew has been previously solved [7, 8, 11, 12]. However, in these papers, clock delays rather than clock skews are calculated and therefore these clock delays cannot be directly used for determining the permissible range of the local clock skews, as described in this paper.

The difference between the propagation delays of a local data path L_{ij} defines the minimum clock period necessary to safely latch data within L_{ij}. The largest difference among all the local data paths of the circuit defines a clock period that can be used to safely latch data into any local data path in the circuit. However, as shown in the example of Figure 3, in the presence of feedback and/or parallel paths, local timing constraints may not be sufficient to determine the minimum clock period (since certain global timing constraints such as (7) must also be satisfied). Nevertheless, a clock period always exists that satisfies all the local and global timing constraints of a circuit. This clock period is bounded by two terms, T_{CPmin} and T_{CPmax}, as demonstrated by Deokar and Sapatnekar in [8]. The definition of the upper and lower bound of the clock period is formally stated below:

Lemma 5: Given a synchronous circuit C modeled by a graph $G(V,E)$, there exists a clock period T_{CP} satisfying all the local and global timing constraints of the circuit, bounded by $T_{CPmin} \le T_{CP} \le T_{CPmax}$, where T_{CPmin} is the largest propagation delay difference of any local data path $L_{ij} \in G$,

$$T_{CP\min} = MAX(\underset{v_{ij}\in V}{MIN}(T_{PD\max ij} - T_{PD\min ij}), \underset{v_i \in V}{MAX}(T_{PD\max ii})) \quad , \quad (12)$$

and T_{CPmax} is the largest propagation delay of any local data path $L_{ij} \in G$,

$$T_{CP\max} = MAX(\underset{v_{ij}\in V}{MAX}(T_{PD\max ij}), \underset{v_i \in V}{MAX}(T_{PD\max ii})) \quad . \quad (13)$$

The second term in (12) and (13) accounts for the self-loop circuit when the output of a register is connected to its input through an optional logic block. Since the initial and final registers are the same, the clock skew in a self-loop is zero and the clock period is determined by the maximum propagation delay of the path connecting the output of the register to its input. Observe that a clock period equal to the lower bound exists for circuits without parallel and/or feedback paths. Furthermore, a clock period equal to the upper bound always exists since the permissible range of any local data path in the circuit contains the zero clock skew value. Although (13) satisfies any local and global timing constraints of circuit C, it is possible to determine a lower clock period that satisfies (7).

Several algorithms for determining the optimal clock period while exploiting non-zero clock skew exist. Fishburn [7] introduced this approach with a linear programming-based algorithm that minimizes the clock period while determining a set of clock path delays to drive the individual registers within the circuit. In [8], Deokar and Sapatnekar present a graph-based approach to achieve a similar goal, followed by an optimization step to reduce the skew between registers while preserving the minimum clock period. Other works, such as Sakallah *et al.* [11] and Szymanski [12], also calculate the optimal clock period and clock path delay schedule using linear programming techniques.

The algorithm used in this work also utilizes a graph-based approach to find the minimum clock period while ensuring that all the permissible ranges in the circuit satisfy (7). The initial clock period is given by (12) and, the local and global permissible ranges for each local data path in C are calculated assuming this clock period. If at least one data path does not satisfy (7), the clock period is increased and the permissible ranges are re-calculated. This iterative process continues until (7) is satisfied for all global data paths. The primary distinction of this algorithm is that the permissible range of each local data path $Pl(L_{ij})$ is determined rather than the clock path delays and is constrained to a sub-set of values, as explained in Section 2.3. This information is crucial for maximizing the performance of a synchronous circuit while considering the effects of process parameter variations in the design of clock distribution networks.

2.3 Selecting Clock Skew Values

Given any two vertices v_k, $v_l \in V$, the set of valid clock skew values between v_k and v_l is given by (7) and bounded by (8) and (9), as described in section 2.2. In the presence of feedback and/or parallel paths, the resulting permissible range $Pg(P_{kl})$ is a sub-set of the permissible range of each independent global data path between v_k and v_l, as exemplified in Figure 3. However, due to (5), $Pg(P_{kl})$ is the sum of the permissible range of each local data path for every global data path P_{kl} connecting v_k and v_l. Therefore, it is necessary to constrain the permissible range of each local data path to a sub-set of values within its original permissible range. Alternatively, if $Pl(L_{ij})$ is the permissible range of a local data path within one of the global data paths connecting v_k and v_l, $\rho(L_{ij})$ is a sub-set of values within $Pl(L_{ij})$ such that $\rho(L_{ij}) \subseteq Pl(L_{ij})$. This new region $\rho(L_{ij})$ is described as the *effective permissible range* of a local data path.

An example of an effective permissible range is the parallel path shown in Figure 3a. For T_{CP} = 8 tu, the permissible range $Pg(P_{13})$ = (-2, -2). Since $Pg(P_{13}) = Pl(L_{12}) + Pl(L_{23})$, the local data paths L_{12} and L_{23} can only assume clock skew values for which the sum is within (-2,-2). In this case, the permissible range of each local data path is reduced to a single value, or $Pl(L_{12})$ = (1,1) and $Pl(L_{23})$ = (-3,-3), respectively.

Assume that the clock period of the circuit in Figure 3a is now increased from 8 tu to 9 tu. The new permissible range $Pg(P_{13})$ = (-2,0) and the effective permissible range of each local data path is $\rho(L_{12})$ = (1,2), $\rho(L_{23})$ = (-3,-2), and $\rho(L_{13})$ = (-2,0), respectively. Note that selecting a clock skew value outside the effective permissible range of a local data path may lead to a race condition since (7) is violated. Also, there is no unique solution to the selection of an effective permissible range unless $\rho(L_{ij}) \doteq Pl(L_{ij})$. For example, $Pl(L_{12})$ could be set to (0,2) and $Pl(L_{23})$ set to (-2,-2), giving the same permissible range $Pg(P_{13})$ = (-2,0). Therefore, given any two vertices v_k, $v_l \in V$ with feedback and/or parallel paths connecting v_k and v_l, the selection of a clock skew schedule requires determining the effective permissible range $\rho(L_{ij})$ for each local data path between v_k and v_l, and the relative position of $\rho(L_{ij})$ within $Pl(L_{ij})$.

The effective permissible range of a local data path $\rho(L_{ij})$ may not be unique, leading to multiple solutions to the clock skew scheduling problem. It is, however, possible to obtain one solution that is most suitable for minimizing the clock period while reducing the possibility of race conditions due to the effects of process parameter variations. This solution for $\rho(L_{ij})$ is derived from the observation that the bounds of the permissible range of any two vertices v_k, $v_l \in V$ (with possible feedback and/or parallel paths connecting v_k and v_l) are maximum when determined by (8) and (9), and that the permissible $Pg(P_{kl})$ bounded by (8) and (9) is unique. Therefore, the clock skew scheduling problem can be divided into two phases. In the first phase, the permissible range of each global data path is derived from (7), with bounds given by (8) and (9). In the second phase, the clock skew schedule is solved by the following process: 1) the permissible range of a global data path $Pg(P_{kl})$ is divided equally among each local data path belonging to each global data path connecting the vertices v_k and v_l; 2) within each global data path each effective permissible range $\rho(L_{ij})$ is placed as close as possible to the upper bound of the original permissible range $Pl(L_{ij})$, thereby minimizing the likelihood of creating any race conditions; and 3) the specific value of the clock skew is chosen in the middle of the effective permissible range, since no prior information describing the variation of a particular clock skew value may exist. From this clock skew schedule, the minimum clock paths delays are determined.

Providing independent clock path delays for each register is impractical due to the large capacitive load placed on the clock source and the inefficient use of die area. A tree structured clock distribution network is more appropriate, where the branching points are selected according to the delay of each clock path, the relative physical position of the clocked registers, and the sensitivity of each local data path to delay variations. Such an approach for determining the structural topology of a clock distribution network is described in the following section.

3. Clock Tree Topological Design

The topology of a clock tree derived from a clock skew schedule must ensure that the clock path delays are accurately implemented while considering the effects of process parameter variations. A tree-structured topology can be based on the hierarchical description of the circuit netlist, on implementing a balanced tree with a fixed number of branching levels from the clock source to each register with a pre-defined number of branching points per node (an example of this approach is a binary tree with n levels for 2^n registers with two branching points per node), on reducing the effects of process parameter variations by driving common local data paths by the same sub-tree, or by implementing each clock path delay with pre-defined delay segments such that the layout area of the clock tree is reduced.

The topology of the clock distribution tree is built by driving common local data paths by the same sub-tree and by assigning precise delay values to each branch of the clock tree such that the skew assignment is satisfied [13]. For this purpose, each clock path delay is partitioned into a series of branches, each branch emulating a precise quantified delay value. Between any two segments, there is a branching point to other registers or sub-trees of the clock tree, where several branches with pre-defined delays are cascaded to provide the appropriate final delay at each leaf node. The selection of the branch delay is dependent upon the minimum propagation delay that can be implemented for a particular fabrication process and the inverter transconductance (or gain). An example of the topology of a clock tree is shown in Figure 4, where the numbers in brackets are the delays assigned to each branch and the numbers in parenthesis are the clock skew assignment.

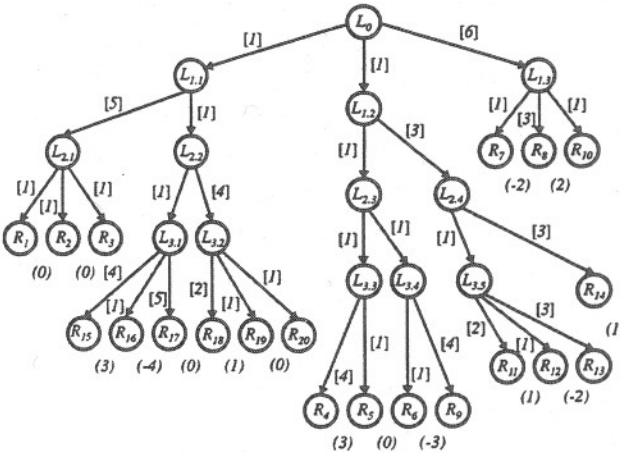


Figure 4: Topology of the clock distribution network

4. Circuit Design of the Clock Tree

The circuit structures are designed to emulate the delay values associated with each branch of the clock tree. Special attention is placed on guaranteeing that the clock skew *between* any two clock paths is satisfied rather than satisfying each individual clock path delay. The successful design of each clock path is primarily dependent on two factors: 1) isolating each branch delay using active elements, specifically CMOS inverters, and 2) using repeaters to integrate the inverter and interconnect delay equations so as to more accurately calculate the delay of each clock path.

The interconnect lines are modeled as purely capacitive lines by inserting inverting buffer repeaters into the clock path such that the output impedance of each inverter is significantly greater than the resistance of the driven interconnect line [14]. As a consequence, the slope of the input signal of a buffer connected to a branching point is identical to the slope of the output signal of the buffer driving that same branching point [15].

In the existing design methodology [9,15], the delay of a branch is implemented with one or more CMOS inverters, as illustrated in Figure 5. The delay equations of each inverter are based on the MOSFET α-power law I-V model developed by Sakurai and Newton [16].

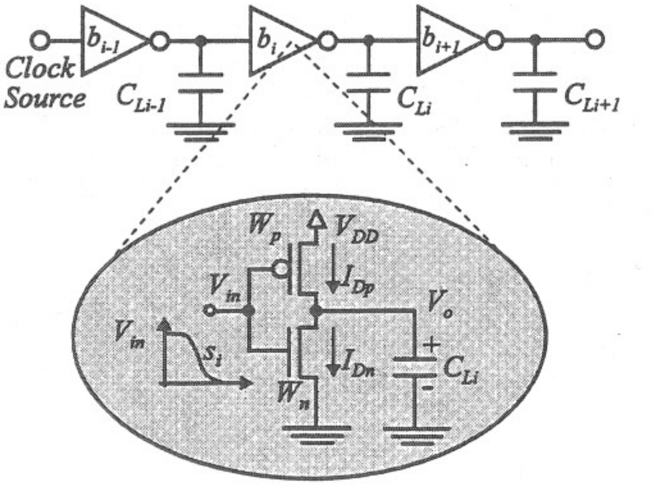


Figure 5: Design of a branch delay element

Each inverter is assumed to be driven by a ramp signal with symmetric rising and falling slopes, selected such that during discharge (charge), the effects of the PMOS (NMOS) transistor can be neglected. The capacitive load of an inverter so as to satisfy a specific branch delay t_{di} is

$$C_{Li} = \frac{2I_{DO}}{V_{DD}}\left[t_{di} - \left(\frac{1}{2} - \frac{1-\nu_{\tau}}{1+\alpha}\right)t_{Ti\text{-}1}\right] \quad , \qquad (14)$$

where I_{DO} is the drain current at $V_{GS} = V_{DS} = V_{DD}$, V_{DO} is the drain saturation voltage at $V_{GS} = V_{DD}$, V_{th} is the threshold voltage, α is the velocity saturation index, V_{DD} is the power supply, t_{di} is the delay of an inverter defined at the 50% V_{DD} point of the input waveform to the 50% V_{DD} point of the output waveform, $\nu_T = V_{th} / V_{DD}$, and t_{Ti} is the transition time of the input signal. Note that C_{Li} is composed of the capacitance of the driven interconnect line and the total gate capacitance of all b_{i+1} inverters. Since t_{di} is known, the only unknown in (14) is the transition time of the input signal t_{Ti} [provided by (15)]. t_{di} can be approximated by a ramp shaped waveform, or by linearly connecting the points 0.1 V_{DD} and 0.9 V_{DD} of the output waveform. This assumption is accurate as long as the interconnect resistance is negligible as compared with the inverter output impedance.

$$t_{Ti} = \frac{t_{0.9} - t_{0.8}}{0.8} = \frac{C_{Li}V_{DD}}{I_{DO}}\left(\frac{0.9}{0.8} + \frac{V_{DO}}{0.8V_{DD}}\ln\frac{10V_{DO}}{eV_{DD}}\right) \quad . \qquad (15)$$

For each clock path within the clock tree, the procedure to design the CMOS inverters is as follows: 1) the load of the initial trunk of the clock tree is determined from (14), assuming a step input clock signal; 2) the slope of the output signal is calculated from (15) and applied in (14) to determine the capacitive load of the following branch, permitting the slope of the output signal to be calculated; and 3) step 2 is repeated for each subsequent branch of the clock path. Steps 1-3 are applied to the remaining clock paths within the clock tree. Observe that if the transition time of the output signal of branch b_i does not satisfy

$$t_{Ti} \leq \frac{1}{\left(\frac{1}{2} - \frac{1-n_T}{1+a}\right)}\left(t_{di+1} - \frac{V_{DD}C_{Li+1}}{2I_{DO}}\right) \quad , \qquad (16)$$

(14) is no longer valid. The transition time t_{Ti} can be reduced in order to satisfy (16) by increasing the output current drive of the inverter in branch b_i. However, increasing I_{DOi} would increase the capacitive load C_{Li} in order to maintain the propagation delay t_{di} for branch b_i. Therefore, the transition time associated with branch b_i must be maintained constant as long as the propagation delay t_{di} of the branch b_i remains the same. Furthermore, the number of inverters required to implement the propagation delay t_{di} is chosen such that (16) is satisfied and the polarity of the clock signal driving branch b_{i+1} does not change.

5. Reduced Sensitivity to Process Parameter Variations

Every semiconductor fabrication process can be characterized by variations in process parameters. These process parameter variations along with environmental variations, such as temperature, supply voltage, and radiation, may compromise both the performance and the reliability of the clock distribution network. A bottom-up approach is presented in this section for verifying the selected clock skew values and correcting for any variations of the clock skew due to process parameter variations that violate the bounds of the permissible range.

Circuit design considerations

Each clock path delay can be modeled as being composed of both a deterministic delay component and a probabilistic delay component. While the deterministic component can be characterized with well developed delay models [e.g., 16], the probabilistic component of the clock path delay is dependent upon variations of the fabrication process and the environmental conditions. The variations of the fabrication process affect both the active device parameters (e.g., I_{DO}, V_{th}, μ_o) and the passive geometric parameters (e.g., the interconnect width and spacing).

The probabilistic delay component is determined for each clock path by assuming that the cumulative effects of the device parameter variations, such as threshold voltage and channel mobility, can be collected into a single parameter characterizing the gain of the inverter, specifically the output current of a CMOS inverter I_{DO} [16]. The minimum and maximum clock path delays are calculated considering the minimum and maximum I_{DO} of each inverter within a branch of the clock distribution network. The worst case variation of the clock skews is determined from the minimum and maximum clock path delays of each local data path. If at least one worst case clock skew value is outside the effective permissible range of the corresponding local data path (i.e., $T_{Skewij} \not\subset \rho(L_{ij})$), a timing constraint is violated and the circuit will not work properly, as illustrated in the example shown in Figure 6.

This violation is passed to the top-down synthesis system, indicating which bound of the effective permissible range is violated. The clock skew of at least one local data path L_{ij} within the system may violate the upper bound of $\rho(L_{ij})$, i.e., $T_{Skewij} > T_{Skewij(max)}$. Observe that if $\rho(L_{ij}) = Pl(L_{ij})$, T_{Skewij} does not satisfy (4), shown as region C in Figure 2, causing zero clocking [7]. By increasing the clock period T_{CP}, the effective permissible clock skew range for each local data path is also increased ($T_{Skewij(max)}$ is increased), permitting those local data paths previously in region C to satisfy (4). The new clock skew value may also violate the lower bound of a local data path, i.e., $T_{Skewij} < T_{Skewij(min)}$, where $T_{Skewij(min)} \subset \rho(L_{ij})$. Observe that if $\rho(L_{ij}) = Pl(L_{ij})$, T_{Skewij} does not satisfy (3), shown as region A in Figure 2, causing double clocking [7]. This situation can be potentially dangerous since the lower bound of $Pl(L_{ij})$ is independent of the clock frequency, causing the circuit to function improperly.

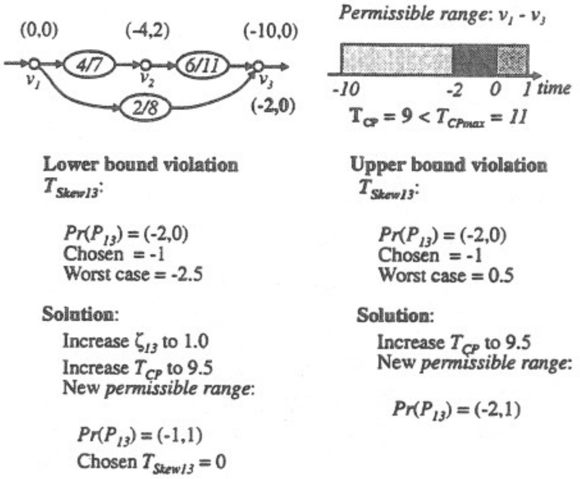


Figure 6: Example of upper and lower bound clock skew violations

Two compensation techniques are required to prevent lower bound violations, depending upon where the effective permissible range of a local data path $\rho(L_{ij})$ is located within the absolute permissible range of the local data path, $Pl(L_{ij})$. If the lower bound of $\rho(L_{ij})$ is greater than the lower bound of $Pl(L_{ij})$, the clock period T_{CP} must be increased to eliminate the race condition, ensuring that the new effective permissible range $\rho(L_{ij}) \subseteq Pl(L_{ij})$. If the lower bound of $\rho(L_{ij})$ is equal to the lower bound of $Pl(L_{ij})$, the clock period cannot be increased and instead the safety term ζ_{ij} [shown in (3)] associated with L_{ij} is increased. Changing ζ_{ij} requires a new set of clock skew values and a minimum clock period T_{CP} to be calculated. Observe that by increasing the safety term ζ_{ij}, the clock skews in the circuit are effectively shifted to the right (see Figure 2), minimizing the likelihood of any race conditions. This iterative process continues until the worst case variations of the selected clock skews no longer violate the corresponding effective permissible ranges of each local data path.

6. Simulation Results

The simulation results presented in this section illustrate the performance improvements obtained by exploiting non-zero clock skew. In order to demonstrate these performance improvements, a set of ISCAS-89

Table 1: Performance improvement with non-zero clock skew

circuit	size	T_{CPo}	T_{CPi}	gain (%)	T_{CP}	gain (%)
	#register/#gates	$T_{Skewij} = 0$	$T_{Skewij} \neq 0$		$T_{SkewI/O} = 0$	
ex1	20/-	11.0	6.3	43.0	7.2	35.0
s27	7/10	9.2	6.6	28.0	9.2	0.0
s298	23/119	16.2	9.4	42.0	9.4	42.0
s344	35/160	28.4	25.6	9.9	25.6	9.9
s386	20/159	19.8	19.8	0.0	19.8	0.0
s444	30/181	18.6	12.2	34.4	12.2	34.4
s510	32/211	19.8	17.3	13.0	17.3	13.0
s938	67/446	27.0	21.4	20.7	25.0	7.4
s1196	45/529	37.0	30.8	16.8	37.0	0.0
s1512	89/780	53.2	43.2	18.8	53.2	0.0

sequential circuits is chosen as benchmark circuits. The performance results are illustrated in Table 1. The number of registers and gates within the circuit including the I/O registers are shown in Column 2. The upper bound of the clock period assuming zero clock skew T_{CPo} is shown in Column 3. The clock period obtained with intentional clock skew T_{CPi} is shown in Column 4. The resulting performance gain is shown in Column 5. The clock period obtained with the constraint of zero clock skew imposed among the I/O registers is shown in Column 6 while the performance gain with respect to zero I/O skew is shown in Column 7.

The results shown in Table 1 clearly demonstrate reductions of the minimum clock period when intentional clock skew is exploited. The amount of reduction is dependent on the characteristics of each circuit, particularly the differences in propagation delay between each local data path. Note also that by constraining the clock skew of the I/O registers to zero, circuit speed can be improved, although less than if this I/O constraint is not used.

Clock distribution networks which exploit intentional clock skew and are less sensitive to the effects of process parameter variations are depicted in Table 2. The ratio of the minimum clock period assuming zero clock skew T_{CPo} to the intentional clock skew T_{CPi} and the per cent improvement is shown in columns 2 and 3, respectively. The permissible range most susceptible to process parameter variations is illustrated in Column 4. The selected clock skew is shown in column 5. In columns 6 and 7, respectively, the nominal and maximum clock skew are depicted, assuming a 15% variation of the drain current I_{DO} of each inverter. Note that both the nominal and the worst case value of the clock skew are within the permissible range. The per cent variation of clock skew due to the effects of process parameter variations is shown in columns 8 and 9. This result confirms the claim stated previously that variations in clock skew due to process parameter variations can be both tolerated and compensated.

7. Conclusions

The problem of scheduling intentional localized clock skew in order to improve performance and reliability while considering the effects of process parameter variations is examined in this paper. A graph-based approach is presented for determining the minimum clock period and the permissible ranges of each local data path. The process of determining the bounds of these ranges and selecting the clock skew value for each local data path so as to minimize the effects of process parameter variations are described. Rather than placing limits or bounds on the clock skew variations, this approach guarantees that each selected clock skew value is within the permissible range despite worst case variations of the clock skew.

Table 2: Worst case variations in clock skew due to process parameter variations, $I_{DO} = 15\%$

circuit	T_{CPo}/T_{CPi}	gain(%)	permissible	selected	Simulated skew (ns)		Error (%)	
			range	clock skew	nom	worst case	nom	worst case
cdn 1	11/9	18.0	[-8,-2]	-3.0	-3.0	-2.10	0.0	30.0
cdn 2	18/15	17.0	[-6.8, -1.4]	-4.2	-4.1	-3.3	2.4	21.4
cdn 3	27/18	33.0	[-14, 2.3]	1.1	1.14	1.3	3.6	18.2

An integrated top-down, bottom-up approach is presented for synthesizing clock distribution networks that are significantly less sensitive to process parameter variations. In the top-down phase, the clock skew schedule and permissible ranges of each local data path are exploited to permit the maximum variation of the clock skew. In the bottom-up phase, worst case variations of clock skew due to process parameter variations are determined from the specific clock distribution network. Variations are compensated by the proper choice of clock skew for each local data path. Results of optimizing the clock skew schedule of several MCNC/ISCAS-89 benchmark circuits are presented. A schedule of the clock skews to make a clock distribution network less sensitive to process parameter variations is presented for several example networks. An 18% improvement in clock frequency with up to a 30% variation in the nominal clock skew, and a 33% improvement in clock frequency with up to an 18% variation in the nominal clock skew are demonstrated for example circuits.

References

[1] S. Pullela, N. Menezes, J. Omar, L. T. Pillage, "Skew and Delay Optimization for Reliable Buffered Clock Trees," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 556-562, November 1993.

[2] H. B. Bakoglu, J. T. Walker, and J. D. Meindl, "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits," *Proceedings of the IEEE International Conference on Computer Design*, pp. 118-122, October 1986.

[3] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng, "Zero Skew Clock Routing with Minimum Wirelength," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. CAS-39, No. 11, pp. 799-814, November 1992.

[4] R.-S. Tsay, "An Exact Zero-Skew Clock Routing Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. CAD-12, No. 2, pp. 242-249, February 1993.

[5] M. Shoji, "Elimination of Process-Dependent Clock Skew in CMOS VLSI," *IEEE Journal of Solid-State Circuits*, Vol. SC-21, No. 5, pp. 875-880, October 1986.

[6] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and System*, IEEE Press, 1995.

[7] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. C-39, No. 7, pp. 945-951, July 1990.

[8] R. B. Deokar and S. Sapatnekar, "A Graph-theoretic Approach to Clock Skew Optimization," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 407-410, May 1994.

[9] J. L. Neves and E. G. Friedman, "Synthesizing Distributed Buffer Clock Trees for High Performance ASICs," *Proceedings of the IEEE ASIC Conference*, pp. 126-129, September 1994.

[10] J. L. Neves and E. G. Friedman, "Design Methodology for Synthesizing Clock Distribution Networks Exploiting Non-Zero Localized Clock Skew," *IEEE Transactions on VLSI Systems*, 1996 (in press).

[11] K. A. Sakallah, T. N. Mudge, O. A. Olukotun, "*checkTc* and *minTc*: Timing Verification and Optimal Clocking of Synchronous Digital Circuits," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 111-117, June 1990.

[12] T. G. Szymanski, "Computing Optimal Clock Schedules," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 399-404, June 1992.

[13] J. L. Neves and E. G. Friedman, "Topological Design of Clock Distribution Networks Based on Non-Zero Clock Skew Specifications," *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, pp. 461-471, August 1993.

[14] S. Dhar and M. A. Franklin, "Optimum Buffer Circuits for Driving Long Uniform Lines," *IEEE Journal of Solid State Circuits*, Vol. SC-26, No. 1, pp. 32-40, January 1991.

[15] J. L. Neves and E. G. Friedman, "Circuit Synthesis of Clock Distribution Networks Based on Non-Zero Clock Skew," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 4.175-4.178, May 1994.

[16] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid State Circuits*, Vol. SC-25, No. 2, pp. 584-594, April 1990.